UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/736,437	12/15/2003	David B. Kirk	NVDA/P000814	3433

26291 7590 10/06/2005

MOSER, PATTERSON & SHERIDAN L.L.P.
595 SHREWSBURY AVE, STE 100
FIRST FLOOR
SHREWSBURY, NJ 07702

EXAMINER	
HSU, JONI	

ART UNIT	PAPER NUMBER
2671	

DATE MAILED: 10/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/736,437	KIRK ET AL.	
	Examiner	Art Unit	
	Joni Hsu	2671	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,11-16 and 23-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,11-16 and 23-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. In light of Applicant's amendment to the heading of the claims, the objection to the specification is withdrawn.

2. In light of Applicant's amendments to Claim 6 and 16, the rejections under 35 U.S.C. 112 have been withdrawn.

3. Applicant's arguments with respect to claims 1-5, 11-16, and 23-32 have been considered but are moot in view of the new ground(s) of rejection.

4. Applicant's arguments, see pages 7-8, filed July 5, 2005, with respect to the rejection(s) of claim(s) 1-5 and 11-16 under 35 U.S.C. 103 have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Bugnion (US006704925B1).

5. Applicant argues that Duluk 1 (US006288730B1) does not teach that the program instructions are for configuring a fragment processor or delaying a fragment process (page 7).

In reply, the Examiner disagrees that Duluk 1 does not teach delaying a fragment process. Duluk 1 describes that if a conflict is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the

conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 inherently discloses that the program instructions are for delaying a fragment process (Col. 7, lines 40-43).

The Examiner agrees that Duluk 1 does not teach configuring a fragment processor. However, Duluk 2 (US006771264B1) describes that OpenGL defines a set of per-fragment operations (Col. 6, lines 34-37), and these per-fragment operations determine the manner in which the pixels are stored (Col. 6, lines 42-65). Therefore, Duluk 2 describes configuring a fragment processor, and Duluk 2 is combined with Duluk 1.

6. Applicant argues that Duluk 2 does not teach reordering operations to take account of conflict detection. The combination of Duluk 1 and Duluk 2 does not teach or suggest eliminating flush instructions or re-ordering execution of fragment process instructions (page 7).

In reply, the Examiner argues that the teachings of Duluk 1 are combined with the teachings of Duluk 2, and Duluk 1 teaches reordering operations to take account of conflict detection (Col. 14, lines 27-40).

The Examiner agrees that the combination does not teach or suggest eliminating flush instructions. However, new grounds of rejection are made in view of Bugnion, as discussed below.

7. Applicant argues that Baldwin (US005815166A) does not teach anything about resolving conflict issues in a manner relevant to the present application (pages 7-8).

In reply, the Examiner agrees. However, Baldwin was just used for the rejection for Claim 7, and Claim 7 is now cancelled, so Baldwin is no longer relevant.

8. Applicant argues that Swanson (US005421028A) does not teach reading a location in graphics memory without an intervening instruction to flush the fragment processing pipeline based on the check of the conflict detection unit (page 8).

In reply, the Examiner agrees. However, new grounds of rejection are made in view of Bugnion, as discussed below.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

11. Claims 1-5 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) in view of Duluk 2 (US006771264B1) in view of Bugnion (US006704925B1).

12. With regard to Claim 1, Duluk 1 describes an application programming interface (Col. 5, lines 29-31) for a programmable graphics processor (Col. 5, lines 7-17; Col. 11, lines 64-65), comprising one or more program instructions (reorder logic, 2623-0, Figure 13b) within the programmable graphics processor (Col. 7, lines 40-41) to detect a position conflict for a position (Col. 14, lines 12-40). Duluk 1 describes that if a conflict is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 discloses preventing a subsequent access of the position until the position conflict is resolved (Col. 14, lines 12-40). Duluk 1 describes determining if a memory conflict is likely to occur based upon the addresses contained in first level reorder queue (2603; Col. 14, lines 18-21), and therefore inherently includes instructions to read the x, y position. Duluk 1 describes that the address is added to first level reorder queue (2603; Col. 14, lines 21-24), and therefore inherently includes instructions to write the x, y position.

However, the program instructions described by Duluk 1 are for reordering the memory addresses to be accessed (Col. 14, lines 1-19), and Duluk 1 does not explicitly teach that these program instructions are for configuring a fragment processor. However, a related patent, Duluk 2, describes that OpenGL defines a set of per-fragment operations (Col. 6, lines 34-37), and these

per-fragment operations determine the manner in which the pixels are stored (Col. 6, lines 42-65). Therefore, Duluk 2 describes one or more program instructions to configure a fragment processor, and these per-fragment operations determine the manner in which the pixels are stored.

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the device of Duluk 1 so that the program instructions are for configuring a fragment processor as suggested by Duluk 2 because Duluk 2 suggests that this is needed in order to determine the manner in which the pixels are stored (Col. 6, lines 34-37, 42-65), and also to perform various tests on the fragments, such as the ownership test, scissor test, alpha test, color test, stencil test, depth buffer test, to determine what operation needs to be performed on the fragment to modify the pixel in the frame buffer at that location (Col. 7, lines 8-22).

However, Duluk 1 and Duluk 2 do not teach that the instructions further includes one or more instructions to write the x, y position and read the x, y position without an intervening instruction to flush the graphics pipeline. However, Bugnion describes allowing for a selective replacement of translations in the case of conflicts. If a comparison fails, a new translation is generated to replace the stale translation. However, the translation cache need not be flushed, as it suffices to patch the stale translation with a jump to the new translation, thereby guaranteeing that the stale translation would never be again executed (Col. 13, line 62-Col. 14, line 5). Therefore, Bugnion inherently discloses that the instructions further includes one or more instructions to write the x, y position and read the x, y position without an intervening instruction to flush.

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Duluk 2 so that the instructions further includes one or more instructions to write the x, y position and read the x, y position without an intervening instruction to flush the graphics pipeline as suggested by Bugnion because Bugnion suggests that flushing is costly in that it wastes time (Col. 15, lines 21-35; Col. 13, line 62-Col. 14, line 5), and therefore it is advantageous to avoid flushing.

13. With regard to Claim 2, Duluk 1 describes that a program instruction (2623-0, Figure 13b) receives memory addresses, and for each memory address received, conflict detection block (2602) determines if a memory conflict is likely to occur based upon the addresses contained in first level reorder queue (2603) (Col. 14, lines 12-40). In order to determine if a memory conflict is likely to occur, the addresses received by a program instruction must inherently include a source location and a destination location.

14. With regard to Claim 3, Duluk 1 describes that the source location includes a buffer identifier corresponding to one of several buffers (Col. 12, lines 2-23).

15. With regard to Claim 4, Duluk 1 describes that the destination location includes a buffer identifier corresponding to one of several buffers (Col. 12, lines 24-41).

16. With regard to Claim 5, Duluk 1 describes that the destination location contains fragment data including at least one of depth, color, and stencil (Col. 8, lines 19-28; Col. 9, lines 30-45).

17. With regard to Claim 27, Duluk 1 describes that the position comprises a region including a plurality of pixels *(texture tile addresses*, Col. 9, lines 30-45; *tiles have 16x16 pixels*, Col. 5, lines 53-56).

18. Claims 11, 12, 15, 16, 23, 25, 26, and 28-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) in view of Bugnion (US006704925B1).

19. With regard to Claim 11, Duluk 1 describes a fragment program for processing fragment data in a fragment processing pipeline (Col. 5, lines 14-17, 42-67), comprising a fragment program instruction to write a destination location in a buffer (Col. 11, line 64-Col. 12, line 7); and a fragment program instruction to read the destination location in the buffer (Col. 11, lines 64-67; Col. 12, lines 20-23).

However, Duluk 1 does not teach that this is done without an intervening instruction to flush the fragment processing pipeline. However, Bugnion describes allowing for a selective replacement of translations in the case of conflicts. If a comparison fails, a new translation is generated to replace the stale translation. However, the translation cache need not be flushed, as it suffices to patch the stale translation with a jump to the new translation, thereby guaranteeing that the stale translation would never be again executed (Col. 13, line 62-Col. 14, line 5). Therefore Bugnion inherently discloses reading the destination location without an intervening instruction to flush. This would be obvious for the same reasons given in the rejection for Claim 1.

20. With regard to Claim 12, Claim 12 is similar in scope to Claim 4, and therefore is rejected under the same rationale.

21. With regard to Claim 15, Duluk 1 describes fragment program instructions to configure the fragment processing pipeline to perform raster operations (Col. 2, lines 21-25; Col. 6, lines 42-67).

22. With regard to Claim 16, Duluk 1 describes that the raster operations are performed using fragment data represented in a floating-point data format (Col. 2, lines 15-18, 21-25; Col. 8, lines 19-41).

23. With regard to Claim 23, Duluk 1 describes a method for processing fragments in a graphics processor pipeline comprising providing a fragment processing unit within the graphics processor pipeline; receiving a first fragment associated with a position by the fragment processing unit; processing the first fragment associated with the position to obtain a processed first fragment (*texture pipeline, texture unit 1200 receives texture coordinates for individual fragments, generates a texture value for each fragment*, Col. 8, lines 9-14). Duluk 1 describes that if a conflict is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 discloses for receiving a second fragment associated with the

position by the fragment processing unit, interlocking the second fragment in part subject to completion of the processing of the first fragment; writing the processed first fragment to a graphics memory; unlocking the second fragment; and processing the second fragment in the fragment processing unit (Col. 14, lines 24-27; Col. 7, lines 40-43).

However, Duluk 1 does not teach that the second fragment is processed without flushing the pipeline between processing the first and second fragments. However, Bugnion describes allowing for a selective replacement of translations in the case of conflicts. If a comparison fails, a new translation is generated to replace the stale translation. However, the translation cache need not be flushed, as it suffices to patch the stale translation with a jump to the new translation, thereby guaranteeing that the stale translation would never be again executed (Col. 13, line 62-Col. 14, line 5). Therefore, the second memory access is processed without flushing between the first and second memory access. This would be obvious for the same reasons given in the rejection for Claim 1.

24. With regard to Claim 25, Duluk 1 describes specifying the position of the first segment as source data for subsequent processing of fragments (*address is added to first level reorder queue 2603, to allow for conflict checking of subsequently received addresses*, Col. 14, lines 18-24).

25. With regard to Claim 26, Duluk 1 describes that if a conflict is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 discloses that the

interlocking step comprises reaching the source data prior to processing the second fragment to prevent writing to the position (Col. 14, lines 18-27).

26. With regard to Claim 28, Duluk 1 describes that if a conflict is determined, the conflicting address request is sent to the conflict queue (2604, Figure 13b; Col. 14, lines 24-26). Since the address request is in the conflict queue, the processor inherently waits until the address request that it is in conflict with has finished processing. Therefore, Duluk 1 discloses checking a location in graphics memory for the processed first fragment prior to unlocking the second fragment (Col. 14, lines 18-27).

27. With regard to Claim 29, Duluk 1 describes processing the first and either the second or the additional fragments in parallel (Col. 9, lines 20-29).

28. With regard to Claim 30, Duluk 1 describes a programmable graphics processor for execution of program instructions (Col. 1, lines 64-67), comprising a read interface (1210, Figure 2) configured to read data from a graphics memory (1213; Col. 8, lines 59-60); a fragment processing unit configured to receive fragments (Col. 7, lines 40-41), each fragment associated with a position (Col. 8, lines 23-24), and the data from the graphics memory and generate processed fragments (Col. 7, lines 40-43); a conflict detection unit (2602, Figure 13b) configured to selectively store the position associated with each fragment and generate a position conflict status (Col. 14, lines 18-24); a write interface (1210, Figure 2) configured to write the processed fragments to the graphics memory (Col. 8, lines 59-60); and a fragment processing pipeline (Col.

5, lines 14-17; Col. 8, lines 19-27) configured to handle read-after-write hazards during execution of shader programs (Col. 6, lines 61-65) including an instruction to write a location in graphics memory, an instruction to check the location in a conflict detection unit and a subsequent instruction to read a location in graphics memory (Col. 14, lines 18-24).

However, Duluk 1 does not teach that the read is done without an interleaving instruction to flush the fragment processing pipeline based on the check of the conflict detection unit. However, Bugnion describes allowing for a selective replacement of translations in the case of conflicts. If a comparison fails, a new translation is generated to replace the stale translation. However, the translation cache need not be flushed, as it suffices to patch the stale translation with a jump to the new translation, thereby guaranteeing that the stale translation would never be again executed (Col. 13, line 62-Col. 14, line 5). Therefore, Bugnion inherently discloses that the read is done without an interleaving instruction to flush based on the check of the conflict detection unit. This would be obvious for the same reasons given in the rejection for Claim 1.

29. With regard to Claim 31, Duluk 1 describes a programmable graphics processor (Col. 1, lines 64-67) including a data cache storing additional data associated with the location, the conflict detection unit determining if the additional data associated with the location is available (Col. 16, lines 38-41).

30. With regard to Claim 32, Duluk 1 describes a programmable graphics processor (Col. 1, lines 64-67) wherein the location is a region comprising a plurality of pixels *(texture tile addresses*, Col. 9, lines 30-45; *tiles have 16x16 pixels*, Col. 5, lines 53-56).

31. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Bugnion (US006704925B1) in view of Wood (US006204856B1).

Duluk 1 and Bugnion are relied upon for the teachings as discussed above relative to Claim 11.

However, Duluk 1 and Bugnion do not teach fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading. However, Wood describes fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading (Col. 1, lines 22-24; Col. 9, lines 64-67; Col. 12, lines 1-6).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Bugnion to include fragment program instructions to configure the fragment processing pipeline to perform depth buffering prior to shading as suggested by Wood because Wood suggests the advantage of reducing the number of attributes to be calculated (Col. 1, lines 62-64; Col. 11, line 62-Col. 12, line 6).

32. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Bugnion (US006704925B1) in view of Isard (US 20040207623A1).

Duluk 1 and Bugnion are relied upon for the teachings as discussed above relative to Claim 11.

However, Duluk 1 and Bugnion do not teach fragment program instructions to configure the fragment processing pipeline to perform depth peeling. However, Isard describes fragment

program instructions to configure the fragment processing pipeline to perform depth peeling [0017, 0055].

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Bugnion to include fragment program instructions to configure the fragment processing pipeline to perform depth peeling as suggested by Isard because Isard suggests that depth peeling is needed to render shadows cast by transparent objects [0055].

33. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Duluk 1 (US006288730B1) and Bugnion (US006704925B1) in view of Knittel (US006266733B1).

Duluk 1 and Bugnion are relied upon for the teachings as discussed above relative to Claim 23.

However, Duluk 1 and Bugnion do not specifically teach processing one or more additional fragments following processing the first fragment without unlocking the second fragment. However, Knittel describes that cases occasionally arise where the fetching of a mini-block from one bank of one DRAM at the end of one row is followed in quick succession by the fetching of a mini-block from the same bank of the same DRAM module at the start of the next row. To avoid this problem, there is a system for reversing the direction of the read-out of a row of mini-blocks when the next mini-block to be read out would result in a fetch from the same or a conflicting bank of the DRAM memory (Col. 3, line 60-Col. 4, line 2). Since the system is reversing the direction of the read-out, it is inherently preventing the reading of the conflicting block and reading another block on the other end of the row. Therefore, Knittel discloses

processing one or more additional voxel data following processing the first block of voxel data without unlocking the second block of voxel data (Col. 3, line 60-Col. 4, line 2; Col. 3, lines 49-50).

It would have been obvious to one of ordinary skill in this art at the time of invention by applicant to modify the devices of Duluk 1 and Bugnion to include processing one or more additional fragments following processing the first fragment without unlocking the second fragment as suggested by Knittel because Knittel suggests the advantage of avoiding conflicts while avoiding the delay cycles of the prior art (Col. 3, lines 9-25).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joni Hsu whose telephone number is 571-272-7785. The examiner can normally be reached on M-F 8am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JH

Kee M. Tung
Primary Examiner